CN 53-1189/**P** **ISSN** 1672-7673

基于 OpenStack 的天文台站计算节点自动管理研究*

邵 岑[1]，邓 辉[1]，王 锋[1,2]，卫守林[1,2]，梅 盈[2]，
石聪明[1]，梁 波[1]，戴 伟[1,2]，柳翠寅[2]

（1. 昆明理工大学云南省计算机技术应用重点实验室，云南 昆明 650500；2. 中国科学院云南天文台，云南 昆明 650011）

摘要：天文数据处理是天文研究的一个重要环节。随着新一代望远镜功能与观测能力的快速发展，在观测地点构建高性能实时计算平台，快速完成数据的分析与处理是一种趋势。针对明安图射电频谱日像仪和明安图观测站实时数据处理系统的建设要求，系统研究了基于 OpenStack 的本地云实现方法与系统自动管理模式，提出了动态进行计算节点启停的方法，并进行了实际测试。实验表明，该模式完全可以满足天文数据处理的需求，并且比传统的静态分配计算资源的数据处理方法更高效，可以有效地节省能源开销，降低观测成本，对未来天文台站高性能计算平台的建设有一定的参考价值。

关键词：天文数据处理；OpenStack；自动管理

中图分类号：TP272 **文献标识码**：A **文章编号**：1672-7673(2017)02-0243-08

近年来，随着天文数据的迅速膨胀，传统的计算方式已经不能满足处理需求，而分布式计算、虚拟化技术以低成本、高效率等优点成为天文数据处理领域的一个研究热点。在天文观测与后续资料处理过程中，需要对大量的观测数据进行实时处理，这对底层的计算资源有非常高的要求，能否提供稳定高效的基础设施支持直接影响数据处理的效率[1]。

明安图射电频谱日像仪(Mingantu Ultrawide SpEctral Radioheliograph，MUSER)位于内蒙古锡林格勒盟正镶白旗，是我国新一代太阳射电综合孔径望远镜[2]。为确保实时观测，在明安图观测站部署了一套高性能计算平台，但随之出现的问题是现场的观测助手与驻站科学家不一定熟悉计算机系统的维护，出现操作系统与软件故障时无法快速恢复等问题[3]。此外，所有的计算节点随时开机，常年电费等是一笔不菲的开销。事实上，在没有太多计算需求的时候，部分计算机节点完全可以脱机进入低能耗状态。

将云计算引入天文数据处理领域，可以简化整体系统的管理要求，并提高系统的可维护性，但静态的虚拟机分配模式往往造成计算资源利用率低或者资源浪费的问题，而人工干预的分配模式又增加系统的成本[4]。本文针对传统天文数据处理模式中存在的计算资源不足、资源浪费及利用率低等问题，提出一种基于 OpenStack 的自动管理模式。该模式采用虚拟化管理软件 OpenStack 作为基础平台，为天文数据处理提供虚拟计算资源并进行动态管理，可根据系统当前的计算量决定增加或者减少计算资源，将承担较少计算量的计算节点关闭，以提高计算资源的利用率并降低物理机开销，改进传统天文数据处理过程中存在的问题。

1 OpenStack 云平台介绍

OpenStack 是一款由美国国家航空和宇航局和云计算提供商(Rackspace)共同合作开发的开源云平台管理软件。从 2010 年第 1 版诞生至今，OpenStack 经历了 13 个版本的演变，期间有成千上万的社

* 基金项目：国家自然科学基金（U1231205）资助.
收稿日期：2016-06-21；修订日期：2016-07-12
作者简介：邵 岑，男，硕士. 研究方向：计算机应用技术. Email：516913794@qq.com
通讯作者：邓 辉，女，教授. 研究方向：天文技术与方法. Email：dh@cnlab.net

区与公司参与了版本的升级与改进。如今，OpenStack 已经足以向各领域提供一套完整的基础设施服务(IaaS)、平台服务(PaaS)、软件服务(SaaS)解决方案，改善并解决行业中存在的问题[5]。

OpenStack 的 3 大核心组件为 Nova、Glance、Swift。Nova 负责虚拟机管理，包括虚拟机的创建、迁移、删除等工作。Nova 自身又包含许多子服务，如 Nova-API 提供服务接口并接收用户请求；Nova-Scheduler 负责虚拟机调度工作，它可以决定在哪台物理节点上创建虚拟机；Nova-Compute 负责虚拟机创建工作[6]。镜像管理组件 Glance 负责镜像的上传、更新、删除等工作，它可以决定 OpenStack 使用的后端镜像存储设备，如使用 Swift 存储或者使用本地文件系统存储。Swift 组件提供对象存储服务，用户可以在 Swift 中创建自己的容器，存储各种类型的文件，如文本、视频、音频等。Swift 提供的对象存储服务具有强大的扩展性、冗余和持久性，并且通过保存多个副本的方式为用户提供极高的安全性[7]。除了这 3 个核心组件外，OpenStack 还包括 Keystone、Quantum、Cinder、Horizon 等组件，分别提供认证服务、虚拟网络服务、虚拟卷服务、界面服务。各组件关系如图 1。

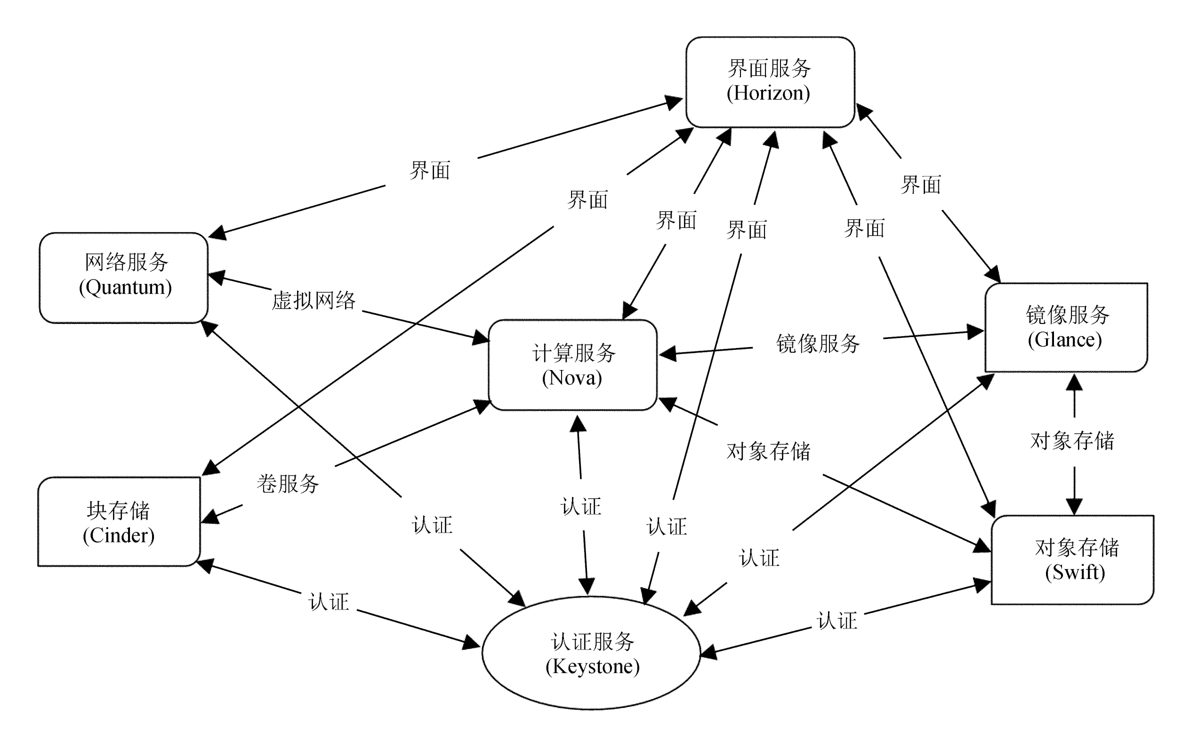


图 1 OpenStack 组件图[8]

Fig. 1 The component diagram of OpenStack[8]

OpenStack 的所有组件既可以结合使用，提供完整的云服务，也可以将每个组件单独安装，提供独立的子服务。通过 OpenStack 的架构，可以看出其功能丰富，有良好的扩展性，并且可以根据需求做大量的二次开发，完全符合大规模天文数据处理的需求。

2 OpenStack 云平台部署

明安图射电频谱日像仪有 8 台服务器，本文采用 1 个控制节点和 N 个计算节点的部署方案，即 1 台服务器做控制节点，剩下的都是计算节点。控制节点负责全局管理，提供身份认证服务、镜像代理服务、虚拟网络服务、客户端网络服务以及数据库服务，数据的存储使用 MySQL 数据库。对于 Swift 和 Nova 来说，控制节点只是提供代理服务，真正实现对象存储和虚拟机创建工作的是计算节点。部署时，所有节点安装 Ubuntu14. 04 Server 版操作系统和 G 版 OpenStack。控制节点上部署 Keystone 服务、Glance 服务、Swift-Proxy 服务、Quantum-Server 服务、Horizon 服务和除 Nova-Compute 外的所有 Nova 服务，所有计算节点部署 Nova-Compute 服务、Quantum-agent 服务、Swift-Storage 服务。节点部署如图 2。

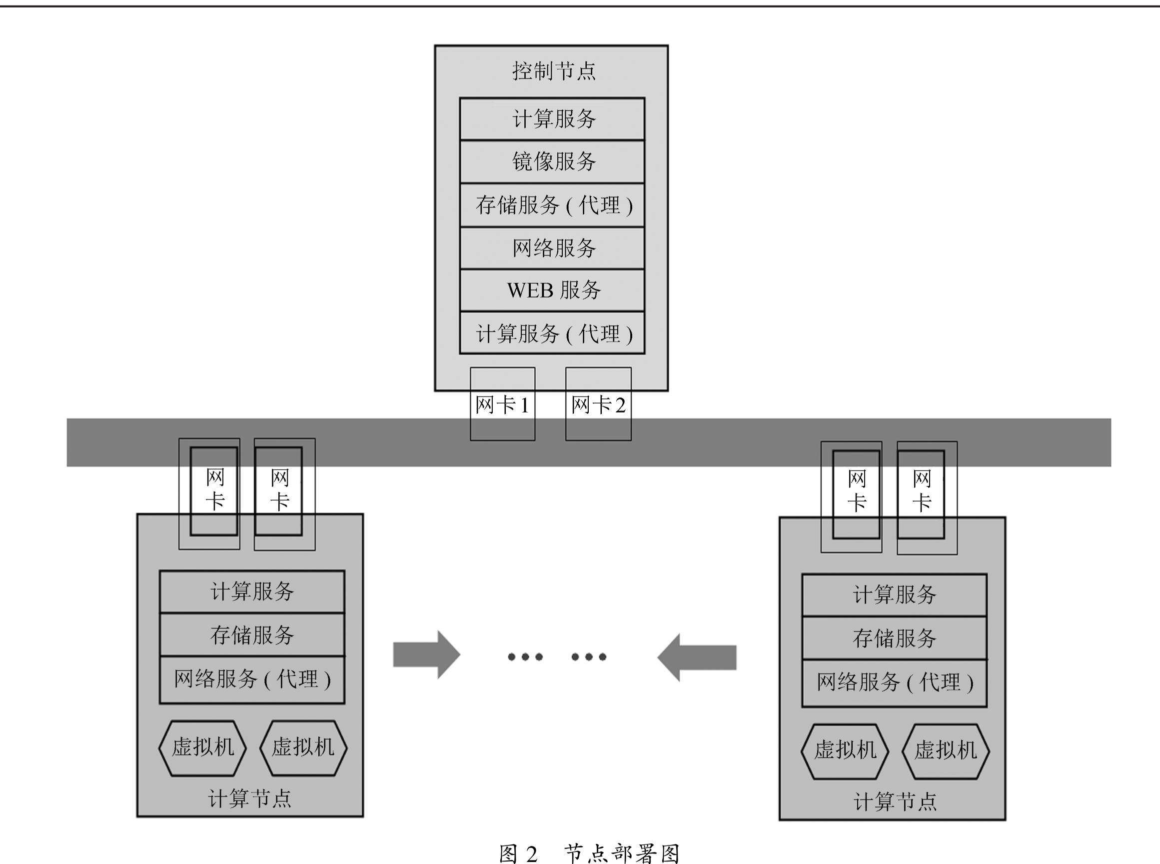


图 2　节点部署图

Fig. 2　The deployment diagram of nodes

控制节点和计算节点分别配有两块网卡，网卡 1 负责节点间通信并提供外网，网卡 2 被 OpenStack 的 Quantum 服务占用，提供虚拟网络服务。本文采用 Open VSwitch 为虚拟机划分 Vlan。为了便于管理，将所有虚拟机先加到一个内部局域网中，确保各个虚拟机之间可以通信，再将内部局域网通过路由器连接到外网，让每台虚拟机可以访问外网资源。这里的外网指用 Quantum 建立的和网卡 1 的 IP 同属一个网段的网络。同时，负责不同功能模块的虚拟机集群可通过划分子网进行网络隔离。网络拓扑图如图 3。为了方便从外网直接访问 OpenStack 中的虚拟机，为每台虚拟机提供了一个浮动 IP，这个浮动 IP 和网卡 1 属同一个网段。

客户端界面由控制节点的 Horizon 服务提供，用户可以直接在浏览器中输入控制节点外网 IP 进行访问。本文的部署方案有很高的扩展性，即计算节点的添加非常方便，只需在新加节点上部署 Nova-Compute 服务、Quantum-agent 服务、Swift-Storage 服务即可。

3　自动管理模式

3.1　设计原理

为实现计算资源的自动管理，监控集群中每台虚拟机的中央处理器使用率、内存使用率和网卡流量，当一台虚拟机在一段时间内各项监控值都低于设定阈值时，说明这台虚拟机承担的计算量较少，则关闭此虚拟机，减少系统开销；当整个集群中的虚拟机在一段时间内各项监控值升高并超过设定阈值时，说明此时的计算量较大，则开启一台虚拟机平衡负载。同时，当某台虚拟机的负载过高时，应及时为其分配更多资源，以适应计算的需求。

3.2　模式设计

为实现计算资源的自动管理模式，关键需要解决两个问题：一是如何获取各计算节点的资源开销情况；二是如何对计算机节点进行启停操作。

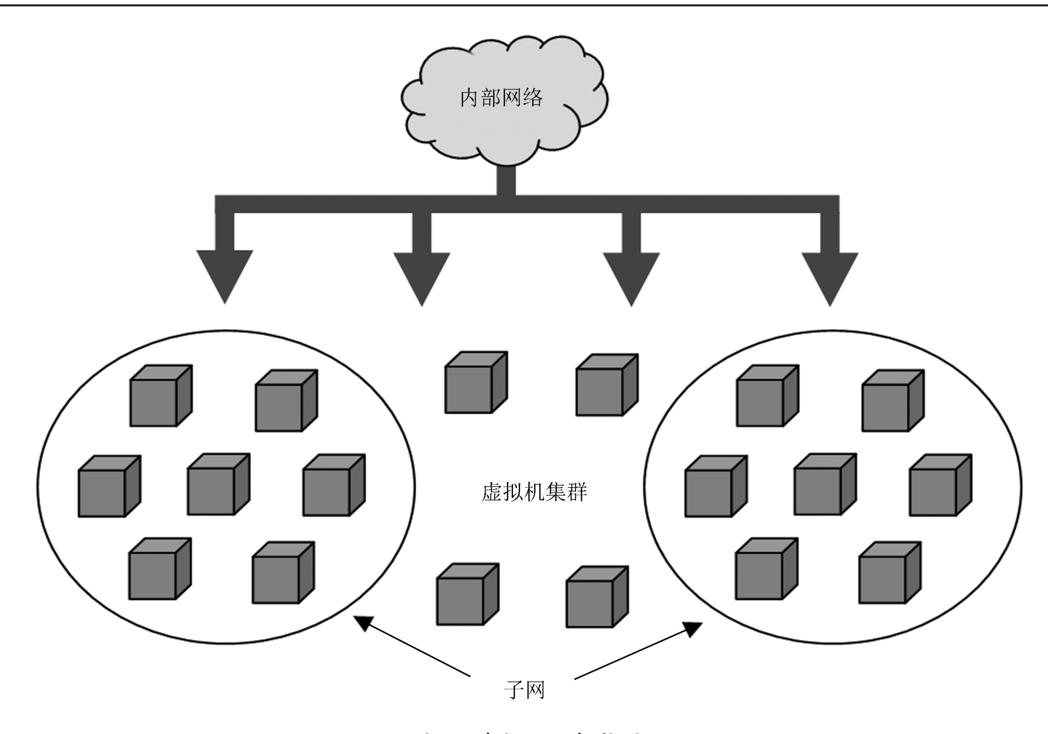


图 3 虚拟网络拓扑图

Fig. 3 Virtual network topology

研究中系统地分析了 OpenStack 的源代码。事实上，OpenStack 的客户端代码提供了一些可调用的应用程序编程接口，但是功能相对分散，而且有些方法所需的参数过多且格式冗长，不便使用。因此需要重新封装一些方法，形成一套便于实现自动管理模式的应用程序编程接口。重新封装的应用程序编程接口有用户管理、镜像管理、虚拟网络管理和虚拟机管理相关的接口。其中几个重要的方法说明如下：

create_user 方法，负责在 OpenStack 中创建用户，需要提供的参数有 Tenant 名、用户名、用户密码。其中 Tenant 代表租户，即 OpenStack 中的一个项目，一个租户下可包含许多用户。此方法首先检查租户是否存在，若不存在则首先创建租户，若存在则在此租户下新建用户，同时检查用户的唯一性并为用户添加相关权限。

create_img 方法，在指定用户下创建一个镜像，需要提供的参数有用户名、用户密码、镜像名、镜像格式和镜像的位置。镜像的位置可以是本地路径或者是一个网络资源的链接。

create_net 方法，在指定用户下创建一个网络，需要的参数有用户名、用户密码、网络名、子网名、网络地址、网关、域名系统、网络地址池和是否支持动态主机配置协议(Dynamic Host Configuration Protocol，DHCP)。

create_vm 方法，在指定用户下创建一个虚拟机，需要的参数有用户名、用户密码、虚拟机名、中央处理器个数、内存大小和镜像名。通过可选参数可以指定虚拟机使用的网络和 IP。

这些方法在创建资源前都会检查用户的合法性和所要创建资源的唯一性。在部署云环境时，镜像的制作是关键，将操作系统、生产环境打包制作成一个镜像，就可以通过这些应用程序编程接口快速地自动部署一个天文数据处理的云环境。当生产环境发生变化时，可通过快照功能将环境保存，再利用快照创建新的虚拟机[9]。

监控集群中虚拟机的方法有很多，采用 OpenStack 底层工具 Libvirt。Libvirt 是 Linux 系统的一套 C 语言编写的函数库，支持多种虚拟化技术的管理，如 KVM、Xen、QEMU 等[10]。Libvirt 命令不会直接提供虚拟机的中央处理器的使用率、内存使用率及网卡流量，但可以根据命令提供的信息计算虚拟机一段时间内各项资源的使用率。主要用到的命令有 virsh vcpuinfo、virsh dommemstat 和 virsh dumpxml，这些命令分别可以返回虚拟机的中央处理器信息、内存信息和启动参数。计算虚拟机中央处理器使用

率时，每隔一段时间对 Libvirt 的 virsh vcpuinfo 命令返回的中央处理器时间参数做一次提取，并记录间隔时间，最后用两个时间点上的中央处理器时间差除以间隔时长得出一段时间内的中央处理器使用率。对于虚拟机的内存使用率，通过计算一段时间内的平均使用率获得。首先将时间段做 N 等分，N 的取值根据时间段的大小而定，利用 Libvirt 的 virsh dommemstat 命令，在此时间段内对该命令返回的瞬时内存使用量 m_i 做 N 次提取，再用这一时刻的瞬时内存使用量 m_i 与虚拟机的总内存 M 做比算出此时的瞬时内存使用率，最后根据(1)式算出这一时段的平均内存使用率。计算虚拟机网卡流量时，利用 Libvirt 的 virsh dumpxml 命令，此命令提供虚拟机的启动信息，从中提取虚拟机使用的网络设备名称，再用 Linux 的 ifconfig 命令得到此设备的流量报告，同样根据一段时间间隔计算虚拟机某时段内上传和下载的流量。

$$U_{\mathrm{AVG}} = \frac{1}{N}\sum_{i=1}^{N}\frac{m_i}{M} \times 100\% . \tag{1}$$

为实现自动管理模式，将资源监控部分封装成一个后台服务，运行在每个计算节点上，服务每隔一段时间获取一次监控值，将数据发送给控制节点。当控制节点收到数据后，将每项数据与设定的安全范围进行比较，若发现虚拟机的平均负载高出安全范围的上限时，则使用 create_vm 方法开启一台新的虚拟机并将其加入到计算集群中，均衡负载；若发现某一台虚拟机的负载低于安全范围的下限时，则使用 delete_vm 方法将其关闭并从计算集群中删除。整个处理流程如图 4。

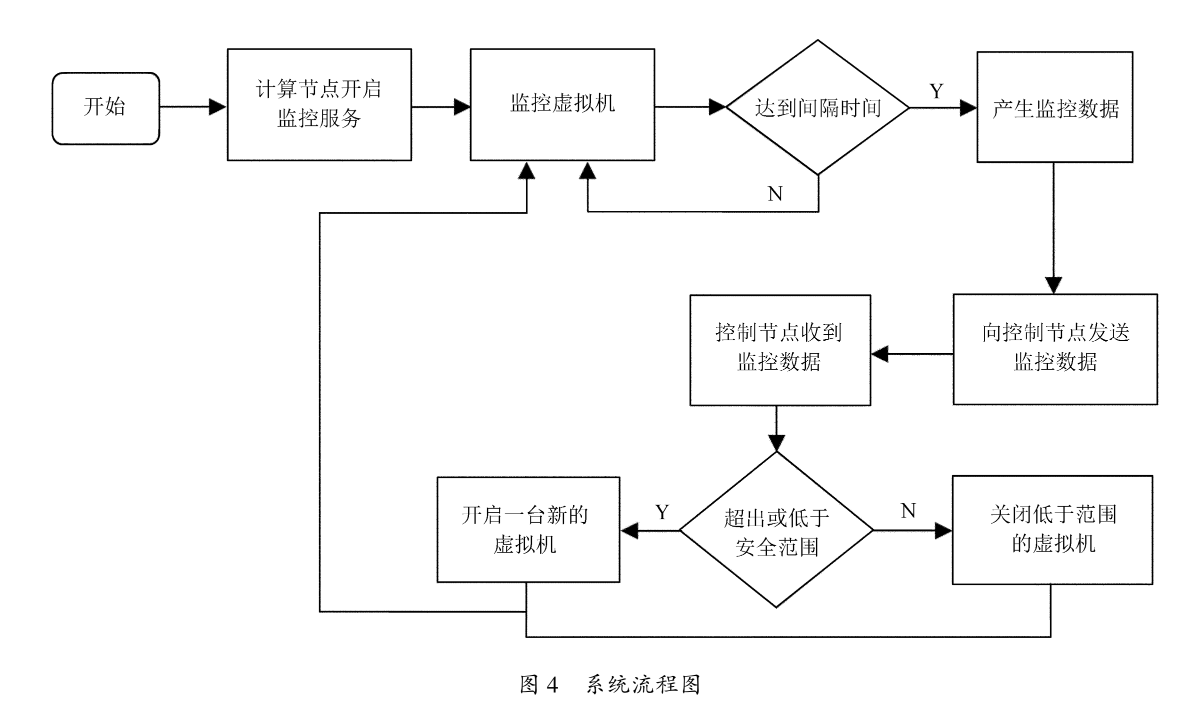


图 4　系统流程图

Fig. 4　System flowchart

4　实验与结论分析

为了验证整体系统的可用性，使用两台刀片服务器部署 OpenStack，其中一台作控制节点，另一台作计算节点。使用封装好的应用程序编程接口在 OpenStack 中自动部署一个用于处理天文观测数据的 Mesos 集群，其中包括租户、用户的创建、镜像的上传、网络的创建和虚拟机的创建。创建网络时需要创建一个内部网络和一个外部网络，所有虚拟机连接到内部网络，便于管理且保证虚拟机间的通信；外部网络除了提供访问外部网络资源的功能外，还可以为虚拟机分配浮动 IP，以便从外部直接访问虚拟机，内网和外网之间使用一个路由器连接。系统初始时有 4 台虚拟机，每台虚拟机配有 1 个虚拟中央处理器和 4 G 内存，内网 IP 的范围为 192. 168. 100. 2-6，浮动 IP 的范围为 172. 31. 254. 81-84。实验环境见表 1。

表 1 实验环境

Table 1 Experiment environment

主机名	VCPU	RAM/GB	内网 IP	外网 IP
Mesos-master	1	4	192. 168. 100. 2	172. 31. 254. 81
Mesos-slave1	1	4	192. 168. 100. 4	172. 31. 254. 82
Mesos-slave2	1	4	192. 168. 100. 5	172. 31. 254. 83
Mesos-slave3	1	4	192. 168. 100. 6	172. 31. 254. 84

根据实际计算任务为系统设定的安全范围是(转换为小数后)：中央处理器使用率 0. 4~0. 8；内存使用率 0. 4~0. 8；网卡流量 0. 4~1(GB)。在计算节点上开启监控服务，监控 Mesos 集群，监控服务每 1 小时向控制节点发送一次监控数据，控制节点在收到数据后与安全范围进行比较，决定开启或关闭虚拟机。

实验开始时，先给 Mesos 集群分配少量计算任务，经过一段时间发现，slave3 和 slave2 处于关闭状态，如图 5；之后逐渐给 Mesos 集群增加计算任务，同样，经过一段时间发现，slave2 和 slave3 重新开启，并且又有一台新的虚拟机 slave4 加入计算集群，如图 6。实验中使用的刀片服务器带有电压传感器，通过查看某时刻的电压值即可算出此时的瞬时功率，由此得到两个时段的功耗情况如图 7。

☐	Instance Name	IP Address	Size	Keypair	Status	Task	Power State
☐	Mesos-slave3	192.168.100.6 172.31.254.84	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Shutoff	None	Shutdown
☐	Mesos-slave2	192.168.100.5 172.31.254.83	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Shutoff	None	Shutdown
☐	Mesos-slave1	192.168.100.4 172.31.254.82	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running
☐	Mesos-master	192.168.100.2 172.31.254.81	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running

图 5 关闭计算资源

Fig. 5 Close the computing resources

☐	Instance Name	IP Address	Size	Keypair	Status	Task	Power State
☐	Mesos-slave4	192.168.100.7 172.31.254.86	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running
☐	Mesos-slave3	192.168.100.6 172.31.254.84	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running
☐	Mesos-slave2	192.168.100.5 172.31.254.83	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running
☐	Mesos-slave1	192.168.100.4 172.31.254.82	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running
☐	Mesos-master	192.168.100.2 172.31.254.81	flavor-mesos | 4GB RAM | 1 VCPU | 10GB Disk	-	Active	None	Running

图 6 添加计算资源

Fig. 6 Add the computing resources

实验表明，本文提出的自动管理模式在计算集群承担任务较多的时候自动增加虚拟机，均衡负载；在计算任务较少的时候关闭一些空闲的虚拟机，有效提高了计算资源的利用率，也为系统减少了不必要的开销。从两种情况的功耗情况来看，运行一小时后，开启 2 台虚拟机时功耗在 2 400~2 500 W 之间，耗电约为 2.4°，而开启 5 台虚拟机时功耗在 2 700~2 800 W 之间，耗电约为 2.7°。两种情况每小时耗电相差约 0.3°，由此可见，在计算任务较少时关闭一些虚拟机可以有效地减少耗电，降低系统成本。这对明安图射电频谱日像仪，对任何天文数据处理的项目，都具有很大的参考价值。

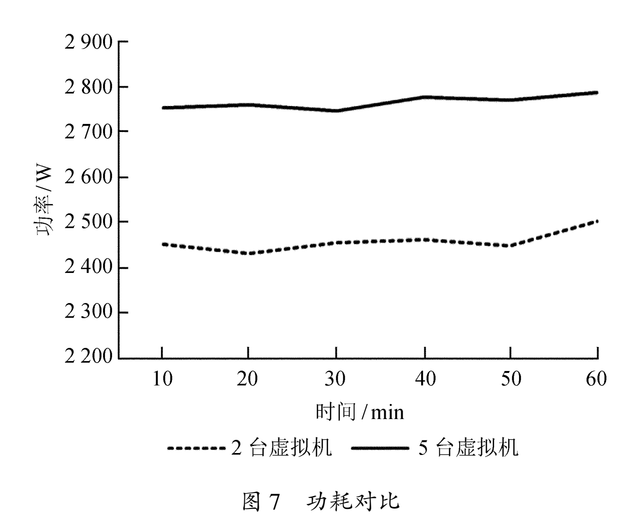


图 7　功耗对比

Fig. 7　The comparison of power consumption

5　结束语

本文就云计算技术在天文数据处理方面的应用进行了研究，介绍了 OpenStack 开源云平台的基本架构和部署方式。以明安图射电频谱日像仪为例，提出了一套基于 OpenStack 的自动管理模式，该模式可应用在大规模的天文数据处理中，根据系统当前计算量动态分配计算资源。与传统的天文数据处理模式相比，本文提出的自动管理模式可有效地提高计算资源的利用率并能为系统减少开销，降低成本。同时，本文测试了基于 OpenStack 的自动管理模式在天文数据处理中的可行性，为天文研究提供参考，具有现实意义。下一步的工作可以研究开发一种自动化程度更高的工具，考虑实现物理节点的自动添加与删除，为天文研究提供更为快捷、高效的数据处理服务。

参考文献：

[1]　卫守林，石聪明，高姣姣，等. Vantage Pro 气象站实时数据采集与在 MUSER 中的应用研究[J]. 天文研究与技术，2016，13(1)：117-123.
Wei Shoulin，Shi Congming，Gao Jiaojiao，et al. A study on the real-time collection of the Vantage Pro weather station and the application in the MUSER [J]. Astronomical Research & Technology，2016，13(1)：117-123.

[2]　周鑫磊，王威，王锋，等. 基于 QT 的 MUSER 观测数据多屏图形化实时显示的设计与实现[J]. 天文研究与技术，2015，12(4)：503-509.
Zhou Xinlei，Wang Wei，Wang Feng，et al. Design and implementation of a multi-monitor display system based on the QT for NAOC MUSER observations [J]. Astronomical Research & Technology，2015，12(4)：503-509.

[3]　颜毅华，张坚，陈志军，等. 关于太阳厘米-分米波段频谱日像仪研究进展 [J]. 天文研究与技术——国家天文台台刊，2006，3(2)：91-98.
Yan Yihua，Zhang Jian，Chen Zhijun，et al. Progress on Chinese solar radioheliograph in cm-dm wavebands [J]. Astronomical Research & Technology——Publications of National Astronomical Observatories of China，2006，3(2)：91-98.

[4]　Liu Yingbo，Wang Feng，Deng Hui，et al. Low-cost high performance distributed data storage for multi-channel observations [J]. New Astronomy，2015，40：78-86.

[5] Rosado T, Bernardino J. An overview of openstack architecture [C] // Proceedings of the 18th International Database Engineering & Applications Symposium. 2014: 366-367.

[6] Beernaert L, Matos M, Vilaca R, et al. Automatic elasticity in OpenStack [C] // Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management. 2012: 1-6.

[7] Hu Bin, Yu Hong. Research of scheduling strategy on OpenStack [C] // Proceedings of the 2013 International Conference on Cloud Computing and Big Data. 2013: 191-196.

[8] Zhang Y, Krishnan R, Sandhu R. Secure information and resource sharing in cloud infrastructure as a service [C] // Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security. 2014: 81-90.

[9] Wang F, Mei Y, Deng H, et al. Distributed data-processing pipeline for mingantu ultrawide spectral radioheliograph [J]. Publications of the Astronomical Society of the Pacific, 2015, 127(950): 383-396.

[10] 梁宇, 杨海波, 李鸿彬, 等. 基于OpenStack资源监控系统 [J]. 计算机系统应用, 2014, 29(4): 44-47.
Liang Yu, Yang Haibo, Li Hongbin, et al. Resource monitoring system based on OpenStack [J]. Computer Systems & Applications, 2014, 29(4): 44-47.

A Study on Automatic Management of Compute Nodes Based on OpenStack for Observatory

Shao Cen[1], Deng Hui[1], Wang Feng[1,2], Wei Shoulin[1,2], Mei Ying[2], Shi Congming[1], Liang Bo[1], Dai Wei[1,2], Liu Cuiyin[2]

(1. Key Laboratory of Applications of Computer Technologies of the Yunnan Province, University of Science and Technology of Kunming, Kunming 650500, China, Email: dh@cnlab.net;
2. Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China)

Abstract: Astronomical data processing is a very important part of astronomical research. With the rapid growth of function and observation capability of the new generation of telescopes, building a high performance real-time computing platform in observing sites and handling the data quickly have become a trend. Based on the real-time data processing system construction requirement of MUSER, this paper has systematically studied the implementation method based upon OpenStack local cloud and automation management pattern. We propose a strategy which can open or close compute nodes automatically and the experiments are carried out. Experiments show that this pattern can meet the needs of astronomical data processing completely and it is more efficient than other traditional data processing methods of static computing resources allocation. At the same time, it can efficiently save our energy bills and reduce the observing costs. This pattern has certain value for reference to build a high-performance computing platform for observatories in the future.

Key words: Astronomical data processing; OpenStack; Automatic management